

CLAIMS:

1. A method comprising:

establishing a secure connection between a network router and a client;

5 initiating a command line interface (CLI) process on the network router;

receiving from the client a CLI command; and

in response to the CLI command, accepting commands encoded in accordance with an extensible markup language.

10 2. The method of claim 1, wherein accepting commands comprises replacing the CLI process with a management server process that provides an extensible markup language-based application programming interface (API) to the client.

15 3. The method of claim 1, further comprising:

receiving from the client, following the replacement of the CLI process, configuration requests and operational requests encoded with extensible markup language tags;

20 accessing a network management interface schema that maps the extensible markup language tags to configuration and operational information associated with software modules running on the network router, the software modules including a chassis software module that defines an inventory of components in the network router chassis, a device configuration software module that defines a physical configuration of the network router, and a routing protocol module that administers protocols supported by the network router;

25 parsing the configuration requests and the operational requests;

accessing the corresponding configuration and operational information associated with the chassis software module, the device configuration software module, and the routing control software module according to the network management interface schema; and

30 emitting replies encoded with extensible markup language tags according to the network management interface schema.

4. The method of claim 3, wherein the tags include chassis tags that define output for the chassis software module, device configuration tags that define output for the

device configuration software module, and routing protocol tags that define output for the routing protocol software module.

5. The method of claim 3, further comprising receiving at least some of the replies encoded with extensible markup language tags from one or more of the software modules.

10 6. The method of claim 3, further comprising encoding at least some of the replies with extensible markup language tags based on non-encoded replies received from one or more of the software modules.

15 7. The method of claim 6, wherein the non-encoded replies include ASCII replies, the method further comprising encoding the ASCII replies in extensible markup language tags to form the encoded replies.

8. The method of claim 3, wherein the extensible markup language tags are defined by one or more data type definition (DTD) files.

20 9. The method of claim 3, wherein the extensible markup language tags are defined by one or more XML Schema Language files.

10. The method of claim 3, wherein the configuration requests include requests to change an existing configuration of the network router, the method further comprising interacting with one or more of the software modules to effect the requested change.

25 11. The method of claim 3, wherein the extensible markup language is XML.

12. The method of claim 3, wherein the network router includes a packet routing engine that specifies one or more routes for data packets, and a packet forwarding engine that forwards the data packets according to the specified routes, and the method is performed in part as a process within an operating environment executing on the packet routing engine.

13. The method of claim 3, further comprising:
receiving from the client, following the replacement of the CLI process, configuration
requests and operational requests encoded with extensible markup language tags;
5 accessing a schema that maps the tags to configuration and operational information
associated with a chassis module, a device configuration module, and a routing protocol
module running on a network router;
accessing the information associated with the software modules; and
emitting replies encoded with extensible markup language tags according to the
10 schema.

14. A computer-readable medium carrying instructions to cause a programmable
processor to:
establish a secure connection between a network router and a client;
15 initiate a command line interface (CLI) process on the network router;
receive from the client a CLI command; and
in response to the CLI command, accept commands encoded in accordance with an
extensible markup language.

20 15. The computer-readable medium of claim 14, wherein the instructions cause
the processor to replace the CLI process with a management server process that provides an
extensible markup language-based application programming interface (API) to the client.

16. The computer-readable medium of claim 14, wherein the instructions cause the processor to:

receive from the client, following the replacement of the CLI process, configuration requests and operational requests encoded with extensible markup language tags;

5 access a network management interface schema that maps the extensible markup language tags to configuration and operational information associated with software modules running on the network router, the software modules including a chassis software module that defines an inventory of components in the network router chassis, a device configuration software module that defines a physical configuration of the network router, and a routing protocol module that administers protocols supported by the network router;

10 parse the configuration requests and the operational requests;

access the corresponding configuration and operational information associated with the chassis software module, the device configuration software module, and the routing control software module according to the network management interface schema; and

15 emit replies encoded with extensible markup language tags according to the network management interface schema.

17. The computer-readable medium of claim 16, wherein the tags include chassis tags that define output for the chassis software module, device configuration tags that define output for the device configuration software module, and routing protocol tags that define output for the routing protocol software module.

20
25 18. The computer-readable medium of claim 16, wherein at least some of the replies encoded with extensible markup language tags from one or more of the software modules.

19. The computer-readable medium of claim 16, wherein the instructions cause the processor to encode at least some of the replies with extensible markup language tags based on non-encoded replies received from one or more of the software modules.

20. The computer-readable medium of claim 16, wherein the non-encoded replies include ASCII replies, the method further comprising encoding the ASCII replies in extensible markup language tags to form the encoded replies.

5 21. The computer-readable medium of claim 16, wherein the extensible markup language tags are defined by one or more data type definition (DTD) files.

22. The computer-readable medium of claim 16, wherein the extensible markup language tags are defined by one or more XML Schema Language files.

10 23. The computer-readable medium of claim 16, wherein the configuration requests include requests to change an existing configuration of the network router, wherein the instructions cause the processor to interact with one or more of the software modules to effect the requested change.

15 24. The computer-readable medium of claim 16, wherein the extensible markup language is XML.

20 25. The computer-readable medium of claim 16, wherein the network router includes a packet routing engine that specifies one or more routes for data packets, and a packet forwarding engine that forwards the data packets according to the specified routes, and wherein the instructions cause the processor to execute a process within an operating environment executing on the packet routing engine.

25

26. The computer-readable medium of claim 14, further comprising:
receiving from the client, following the replacement of the CLI process, configuration
requests and operational requests encoded with extensible markup language tags;
accessing a schema that maps the tags to configuration and operational information
5 associated with a chassis module, a device configuration module, and a routing protocol
module running on a network router;
accessing the information associated with the software modules; and
emitting replies encoded with extensible markup language tags according to the
schema.

10

27. A network router management interface comprising:
a secure protocol module that provides a secure connection between a network router
and a client;
a command line interface (CLI) module that receives CLI commands from a client;
15 and
a management server module that receives the CLI commands from the CLI module
and, in response to one of the CLI commands, accepts commands encoded in accordance
with an extensible markup language.

20

28. The network router management interface of claim 27, wherein the
management server module accepts the commands by replacing the CLI module with the
management server module and providing an extensible markup language-based application
programming interface (API) to the client.

25

29. The network router management interface of claim 27, wherein the
management server module:
receives from the client, following the replacement of the CLI module, configuration
requests and operational requests encoded with extensible markup language tags,
accesses a network management interface schema that maps the extensible markup
30 language tags to configuration and operational information associated with software modules
running on the network router, the software modules including a chassis software module

that defines an inventory of components in the network router chassis, a device configuration software module that defines a physical configuration of the network router, and a routing protocol module that administers protocols supported by the network router,

parses the configuration requests and the operational requests,

5 accesses the corresponding configuration and operational information associated with the chassis software module, the device configuration software module, and the routing control software module according to the network management interface schema, and

emits replies encoded with extensible markup language tags according to the network management interface schema.

10

30. The network router management interface of claim 29, wherein the tags include chassis tags that define output for the chassis software module, device configuration tags that define output for the device configuration software module, and routing protocol tags that define output for the routing protocol software module.

15

31. The network router management interface of claim 30, wherein the management server module receives at least some of the replies encoded with extensible markup language tags from one or more of the software modules.

20

32. The network router management interface of claim 29, wherein the management server module encodes at least some of the replies with extensible markup language tags based on non-encoded replies received from one or more of the software modules.

25

33. The network router management interface of claim 32, wherein the non-encoded replies include ASCII replies, and the management server module encodes the ASCII replies in extensible markup language tags to form the encoded replies.

34. The network router management interface of claim 29, wherein the extensible markup language tags are defined by one or more data type definition (DTD) files.

35. The network router management interface of claim 29, wherein the extensible markup language tags are defined by one or more XML Schema Language files.

36. The network router management interface of claim 29, wherein the configuration requests include requests to change an existing configuration of the network router, the management server module interacting with one or more of the software modules to effect the requested change.

37. The network router management interface of claim 29, wherein the extensible markup language is XML.

38. The network router management interface of claim 29, wherein the network router includes a packet routing engine that specifies one or more routes for data packets, and a packet forwarding engine that forwards the data packets according to the specified routes, and the management server module executes as a process within an operating environment on the packet routing engine.

20